MOTOROLA
Semiconductor Products Inc.

AN-808
Application Note

INTERFACING M6800 PERIPHERAL DEVICES TO THE MC68000 ASYNCHRONOUSLY

Prepared by:
Arnold J. Morales
Microprocessor Applications Engineer
Austin, Texas

This application note describes a technique for interfacing M6800 peripheral devices to a MC68000 microprocessor using a four-chip TTL circuit. Any M6800 peripheral is easily interfaced to the MC68000 using the M6800 peripheral control interface (E, $\overline{VMA}$, $\overline{VPA}$) that is designed into the MC68000. However, when using this interface, the peripheral must be driven by the MC68000 enable (E) signal. The frequency of this clock is one-tenth of the MC68000 clock frequency with a 60/40 (6 clocks high, 4 clocks low) duty cycle. Certain applications may require a clock frequency other than the one-tenth sample that is readily available. An application using a MC68B54 Advanced Data Link Controller (ADLC) at a high data transfer rate could require an E clock frequency of up to two megahertz because the data transfer rate of the ADLC depends on the transmit and receive clocks which are limited by the E clock frequency.

TIMING CONSIDERATIONS

Typical read and write timing for the MC68000 is shown in Figure 1. The relationship between the MC68000 timing and the access timing for the interface circuit given in this application is shown in Figure 2. The best case timing has data strobe occurring with the minimum setup time to allow peripheral selection on the next falling edge of the E clock. In the worst case timing, the data strobe did not occur in time to allow peripheral selection on the next falling edge of E. Therefore, a full E cycle has to occur and then the peripheral selection is done on the falling edge of that full cycle. The resulting cycle times for these best and worst cases and a comparison between asynchronous and synchronous interfacing is summarized in Table 1.

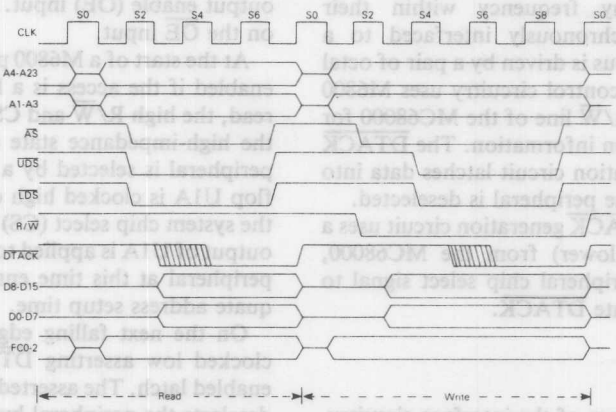

Figure 1. MC68000 Read and Write Cycle Timing Diagram

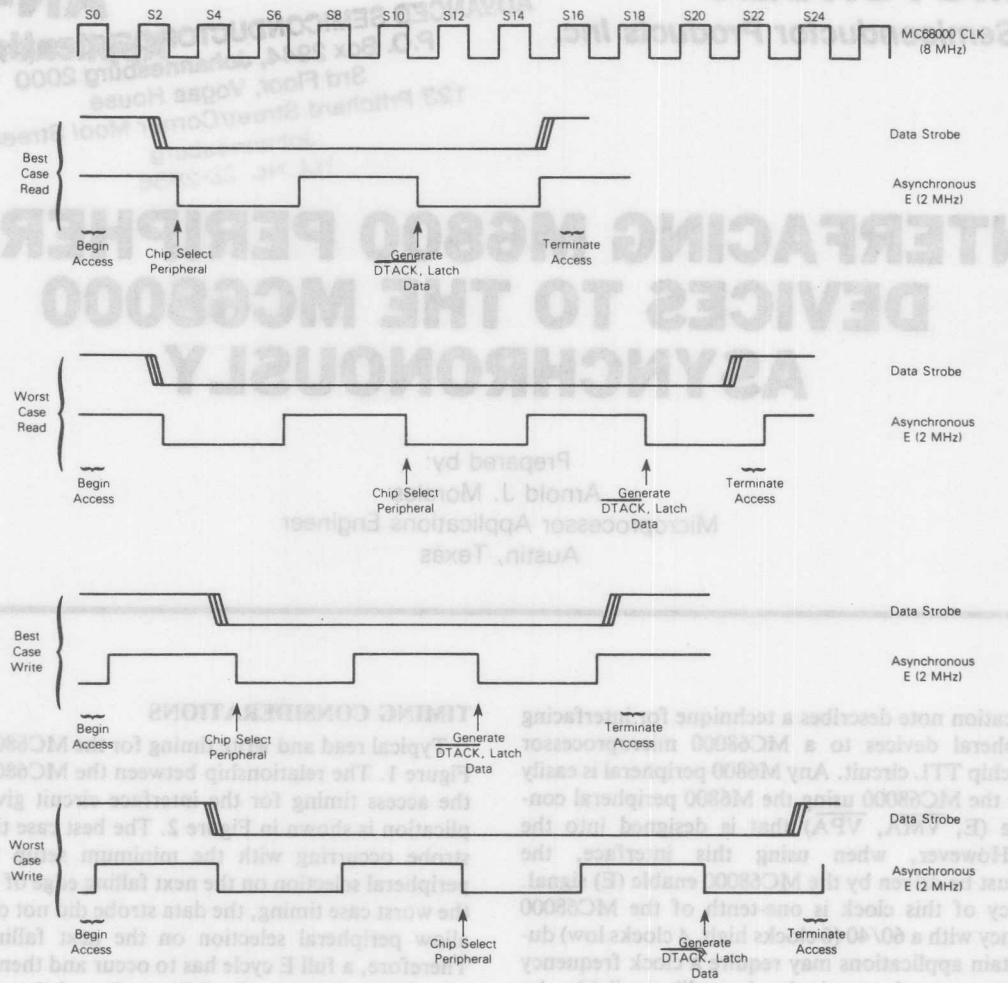

Figure 2. Asynchronous Interface Access Timing Diagrams

BLOCK DIAGRAM

Figure 3 is a block diagram of a circuit that allows M6800 peripherals, operating at any frequency within their operating range, to be asynchronously interfaced to a MC68000 processor. The data bus is driven by a pair of octal transparent latches. The latch control circuitry uses M6800 peripheral chip select and the R/$\overline{W}$ line of the MC68000 for output enable and data direction information. The $\overline{DTACK}$ signal from the $\overline{DTACK}$ generation circuit latches data into the enabled octal latch when the peripheral is deselected.

The peripheral select and $\overline{DTACK}$ generation circuit uses a data strobe (either upper or lower) from the MC68000, peripheral E, and a M6800 peripheral chip select signal to select the peripheral and generate $\overline{DTACK}$.

CIRCUIT OPERATION

Figure 4 is a schematic diagram of the interface circuitry. Refer to this diagram during the following discussion. Initially flip flops U1A and U1B are cleared causing a high $\overline{DTACK}$ output setting U2 and U3 to a transparent mode. Latch U2 is in the high-impedance state due to a high on the output enable ($\overline{OE}$) input. Latch U3 is enabled due to a low on the $\overline{OE}$ input.

At the start of a M6800 peripheral access, latch U3 remains enabled if the access is a MC68000 write. If the access is a read, the high R/$\overline{W}$ and CS inputs to U4A cause U3 to go to the high-impedance state and U2 to become enabled. The peripheral is selected by a low chip select prime ($\overline{CS}$'). Flip flop U1A is clocked high on the first falling edge of E with the system chip select (CS) and data strobe (DS) high. The Q output of U1A is applied to U4D, asserting $\overline{CS}$'. Selecting the peripheral at this time ensures that the peripheral has adequate address setup time.

On the next falling edge of E, the $\overline{Q}$ output of U1B is clocked low asserting $\overline{DTACK}$ and latching data into the enabled latch. The asserted $\overline{DTACK}$ signal, inverted by U4D, deselects the peripheral by causing $\overline{CS}$' to go high. Flip flop U1 is cleared by DS going low when the access terminates. Clearing U1 also initializes the interface circuitry for the next access.

Table 1. Synchronous and Asynchronous Interface Access Time

	Read Access Times (MC68000 Cycles)		Write Access Times (MC68000 Cycles)	
	Best Case	Worst Case	Best Case	Worst Case
Synchronous	9	18	9	18
Asynchronous	8	11	9	12

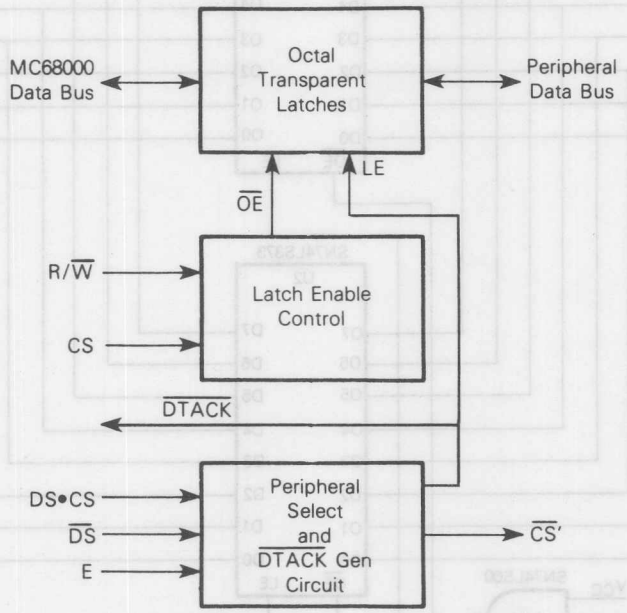

Figure 3. M6800 Peripheral to MC68000 Interface — Block Diagram

SAMPLE CIRCUIT

An example of this interface circuitry is given in the following paragraphs. This example illustrates how the MC68000 can be interfaced to both a MC6854 Advanced Data Link Controller (ADLC) and a MC6840 Programmable Timer Module (PTM) at the same time. The circuit shown in Figure 5 uses the two megahertz "B" version parts connected to a MC68000 driven at eight megahertz.

The base addresses for the peripherals in this example are $18001 for the ADLC and $18801 for the PTM. When the MC68000 transfers bytes it asserts the upper data strobe for even addresses and the lower data strobe for odd addresses. The circuit in this example uses the lower data strobe; therefore only odd MC68000 addresses are used. A memory map of the example system is given in Figure 6.

Device Selection — A SN74LS138 1-of-8 decoder (U5) used as an address decoder is used in conjunction with a chip select signal (CS') developed by U6E to select either the ADLC or the PTM. The PTM requires two chip select inputs, one high and one low, to be selected. The low input is provided by the $\overline{O5}$ output of U5 while the high input is provided by an inverted sample of the $\overline{CS'}$, developed by U6E.

To select the ADLC, the $\overline{O5}$ output of U5 must be high and the $\overline{O1}$ output must be low. The ANDing of $\overline{O5}$ with the high CS' developed by U6E generates the low chip select input required by the ADLC.

Test Program — A flow chart of the test program is given in Figure 7 and a listing is provided in Figure 8. Refer to these figures during the following discussions. The first five lines of code initiate operation of timer 3 in the PTM in the continuous mode, resulting in a square wave at the output of timer 3, pin 6. The remaining lines of code are for testing the ADLC. The test program is based on the loopback test program given in Motorola publication MC6854UM (AD).

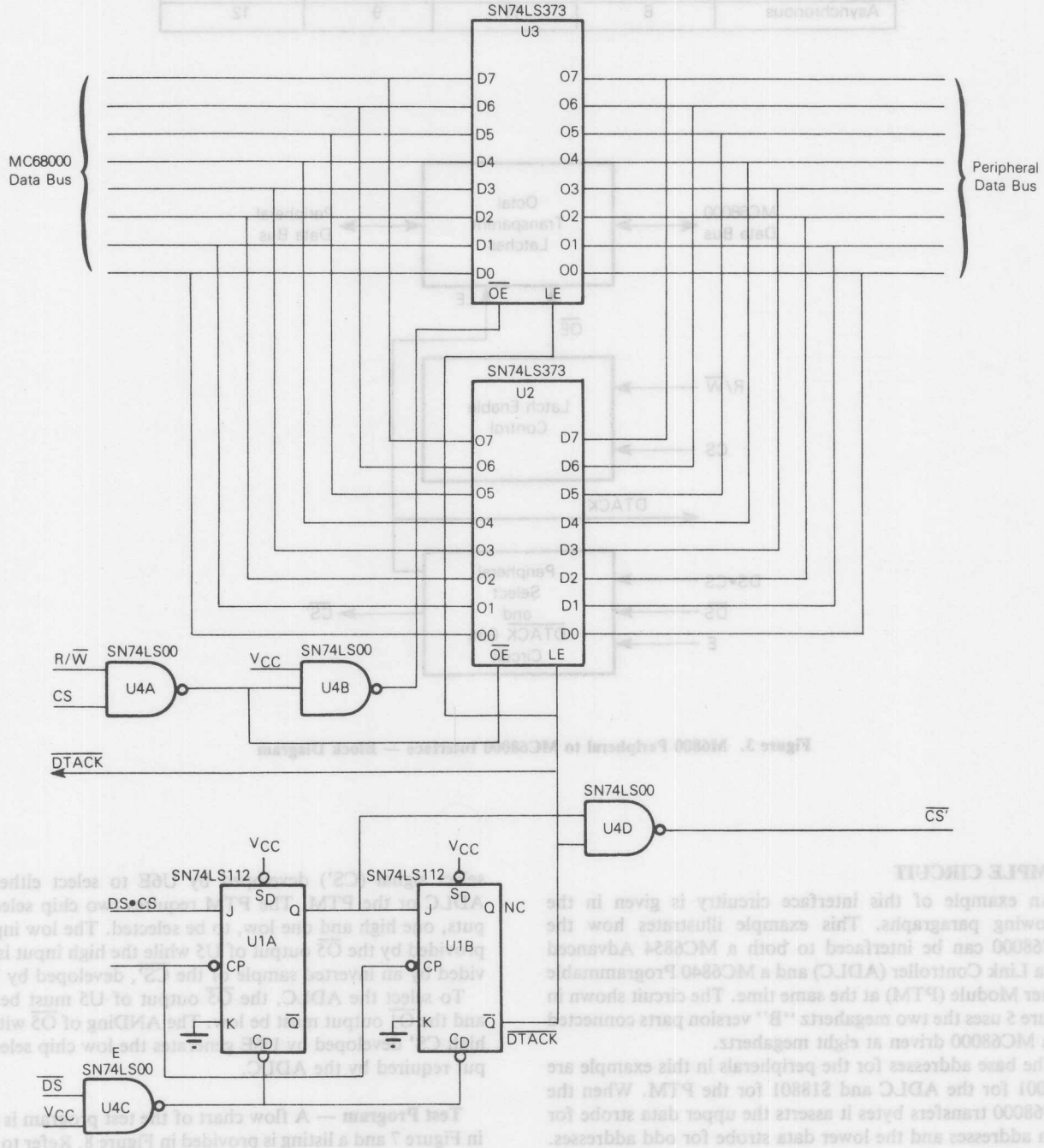

Figure 4. M6800 Peripheral to MC68000 Interface — Schematic Diagram

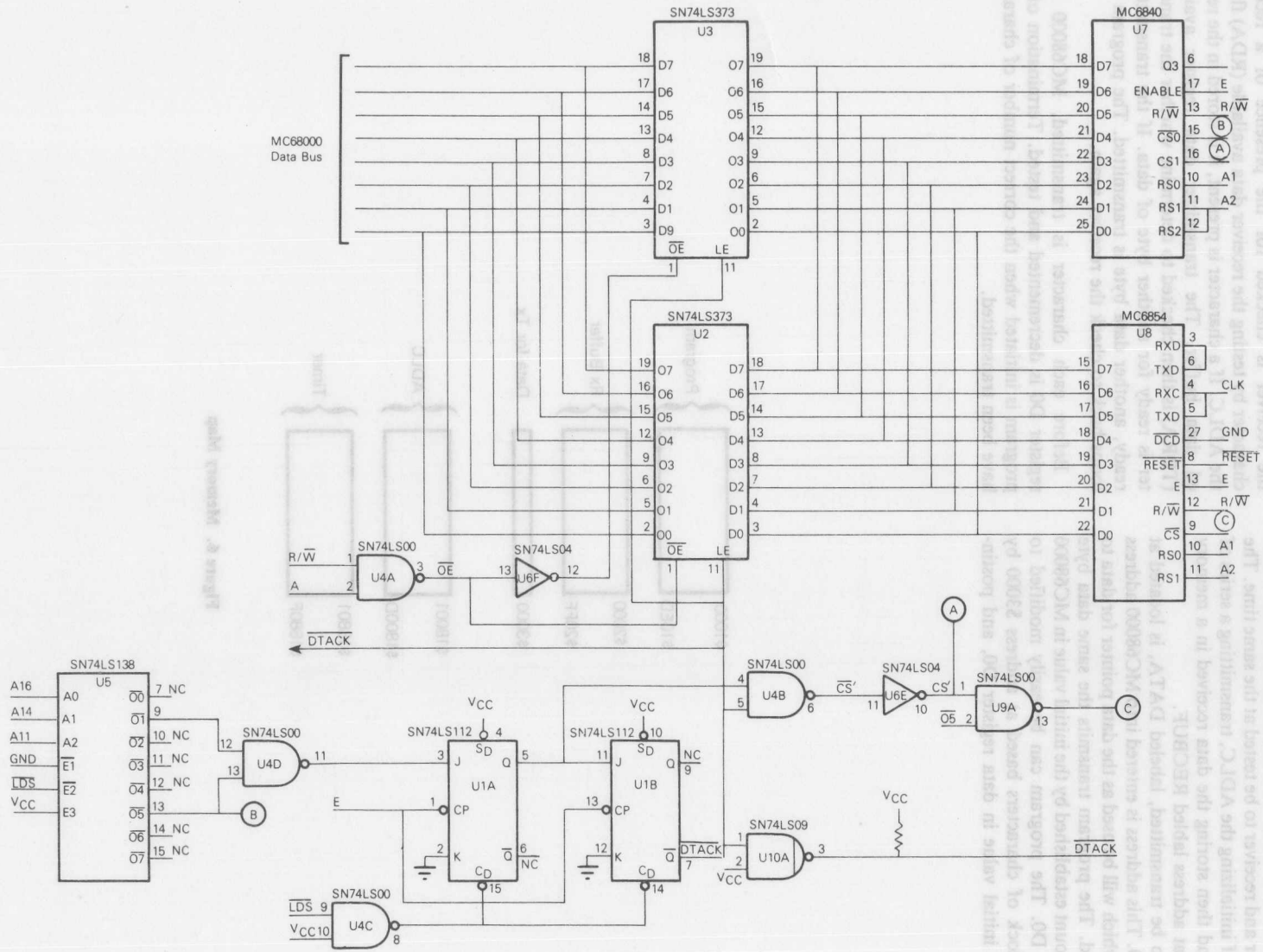

Figure 5. MC68000/MC6840/MC6854 Circuit Example

The ADLC transmitter and receiver clock inputs (TxC, RxC), are tied together and provided with a clock frequency determined by the desired data transfer rate. The transmitter output (TxD) is tied to the receiver input (RxD) to allow both the transmitter and receiver to be tested at the same time. The test consists of initializing the ADLC, transmitting a series of data bytes, and then storing the data received in a memory buffer based at address labled RECBUF.

The byte to be transmitted, labeled DATA, is located at address $3000. This address is entered into MC68000 address register A1, which will be used as the data pointer for data to be transmitted. The program transmits the same data byte 128 times, a count established by the initial value in MC68000 data register D0. The program can be easily modified to transmit a block of characters based at address $3000 by changing the initial value in data register D0, and postincrementing address register A1 after each character is transmitted (line 72).

The main program is a looping, polling sequence. First, the receiver is checked for the presence of a received character by testing the receiver data available (RDA) flag in the ADLC. If a character is present, it is stored in the received data buffer. The transmitter data register available (TDRA) is then checked to determine whether the transmitter is ready for another byte of data. If the transmitter is ready, another data byte is transmitted. The program then loops back to check the receiver again.

Before each character is transmitted, MC68000 data register D0 is decremented and tested. Termination of the program is initiated when the correct number of characters have been transmitted.

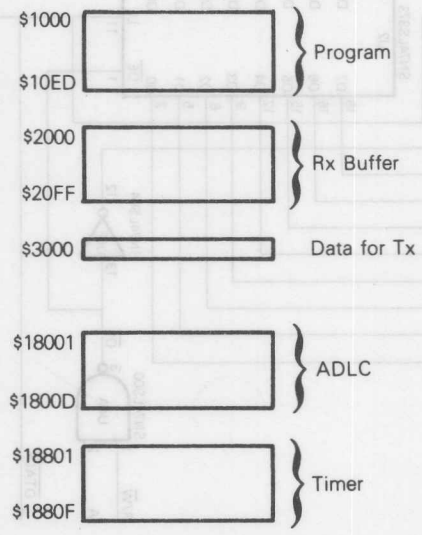

Figure 6. Memory Map

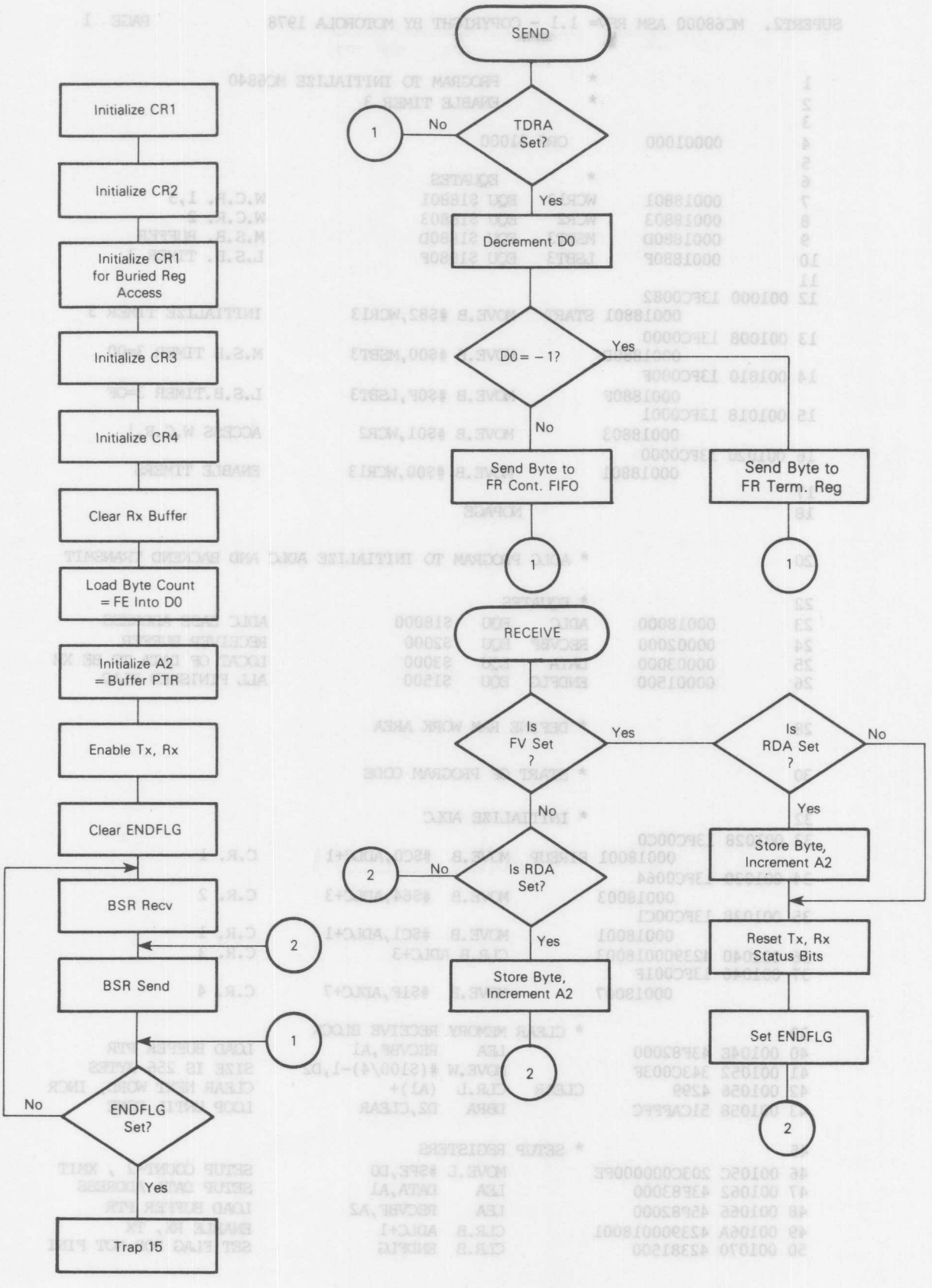

Figure 7. Program Flowchart

```
SUPERT2.  MC68000 ASM REV= 1.1 - COPYRIGHT BY MOTOROLA 1978          PAGE  1

 1                      *         PROGRAM TO INITIALIZE MC6840
 2                      *         ENABLE TIMER 3
 3
 4          00001000          ORG $1000
 5
 6                      *         EQUATES
 7          00018801    WCR13   EQU $18801              W.C.R. 1,3
 8          00018803    WCR2    EQU $18803              W.C.R. 2
 9          0001880D    MSBT3   EQU $1880D              M.S.B. BUFFER
10          0001880F    LSBT3   EQU $1880F              L.S.B. TIMER 3
11
12 001000 13FC0082
               00018801 START   MOVE.B #$82,WCR13       INITIALIZE TIMER 3
13 001008 13FC0000
               0001880D         MOVE.B #$00,MSBT3       M.S.B TIMER 3=00
14 001010 13FC000F
               0001880F         MOVE.B #$0F,LSBT3       L.S.B.TIMER 3=OF
15 001018 13FC0001
               00018803         MOVE.B #$01,WCR2        ACCESS W.C.R.1
16 001020 13FC0000
               00018801         MOVE.B #$00,WCR13       ENABLE TIMERS
17
18                              NOPAGE

20                      * ADLC PROGRAM TO INITIALIZE ADLC AND BACKEND TRANSMIT

22                      * EQUATES
23          00018000    ADLC    EQU   $18000            ADLC BASE ADDRESS
24          00002000    RECVBF  EQU   $2000             RECEIVER BUFFER
25          00003000    DATA    EQU   $3000             LOCAT OF DATA TO BE XM
26          00001500    ENDFLG  EQU   $1500             ALL FINISHED FLAG

28                      * DEFINE RAM WORK AREA

30                      * START OF PROGRAM CODE

32                      * INITIALIZE ADLC
33 001028 13FC00C0
               00018001 FIREUP  MOVE.B  #$C0,ADLC+1     C.R. 1
34 001030 13FC0064
               00018003         MOVE.B  #$64,ADLC+3     C.R. 2
35 001038 13FC00C1
               00018001         MOVE.B  #$C1,ADLC+1     C.R. 1
36 001040 423900018003          CLR.B ADLC+3            C.R. 3
37 001046 13FC001F
               00018007         MOVE.B  #$1F,ADLC+7     C.R. 4

39                      * CLEAR MEMORY RECEIVE BLOCK
40 00104E 43F82000              LEA     RECVBF,A1       LOAD BUFFER PTR
41 001052 343C003F              MOVE.W #($100/4)-1,D2   SIZE IS 256 BYTES
42 001056 4299          CLEAR   CLR.L   (A1)+           CLEAR NEXT WORD, INCR
43 001058 51CAFFFC              DBRA    D2,CLEAR        LOOP UNTIL DONE

45                      * SETUP REGISTERS
46 00105C 203C000000FE          MOVE.L #$FE,D0          SETUP COUNT-2 , XMIT
47 001062 43F83000              LEA     DATA,A1         SETUP DATA ADDRESS
48 001066 45F82000              LEA     RECVBF,A2       LOAD BUFFER PTR
49 00106A 423900018001          CLR.B   ADLC+1          ENABLE RX, TX
50 001070 42381500              CLR.B   ENDFLG          SET FLAG FOR NOT FINI
```

Figure 8. Program Listing

```
52                          * PROCESS TRANSMIT AND RECEIVE TASK TILL DONE
53 001074 61000038          PROCES  BSR     RECV                ATTEMPT RECEIVE
54 001078 6100000C                  BSR     SEND                ATTEMPT TO SEND
55 00107C 4A381500                  TST.B   ENDFLG              FINISHED?
56 001080 67F2                      BEQ     PROCES              LOOP IF NOT FINISHED
57 001082 4E4F                      TRAP    15                  BREAKPOINT WHEN FINIS
58 001084 0000                      DC.W    #0                  BREAKPOINT CODE


60                          * ATTEMPT TO TRANSMIT A CHARACTER
61 001086 08390006
               00018001 SEND        BTST.B #06,ADLC+1          TDRA SET? XMIT READY?
62 00108E 67000014                  BEQ RETURN
63 001092 0C40FFFF                  CMP  #-1,D0                ? LAST BYTE SENT?
64 001096 6700000C                  BEQ    RETURN              YES IGNORE SENDING MO
65 00109A 51C8000A                  DBRA   D0,MORE             COUNT DOWN, BRANCH NO
66                          * PROCESS LAST BYTE BY TERMINATING TRANSMISSION
67 00109E 13D100018007              MOVE.B (A1),ADLC+7         LAST BYTE INTO FRAME
68 0010A4 4E75              RETURN  RTS                        RETURN TO MAINLINE
69                          * PROCESS NEXT BYTE TO TRANSMIT (NOT THE LAST)
70 0010A6 13D100018005      MORE    MOVE.B (A1),ADLC+5         SEND NEXT BYTE TO FRA
71 0010AC 4E75                      RTS                        RETURN TO CALLER


73                          * ATTEMPT TO RECEIVE A CHARACTER
74 0010AE 08390001
               00018003 RECV        BTST.B  #1,ADLC+3          ? FRAME RECEIVED
75 0010B6 66000008                  BNE     GOTFRM             BRANCH IF SO
76 0010BA 61000016                  BSR     TRYINP             ATTEMPT INPUT OF NEXT
77 0010BE 4E75                      RTS                        RETURN TO MAINLINE
78                          * END OF FRAME PROCESSING
79 0010C0 61000010          GOTFRM  BSR     TRYINP             INSURE LAST BYTE PROC
80 0010C4 13FC0064
               00018003             MOVE.B #$64,ADLC+3         CLEAR TX,RX STATUS
81 0010CC 52381500                  ADD.B  #1,ENDFLG           FLAG RECEIVER DONE
82 0010D0 4E75                      RTS                        RETURN TO MAINLINE


84                          * PROCESS INPUT BYTE IF ANY
85 0010D2 08390000
               00018001 TRYINP      BTST.B #0,ADLC+1           ? INPUT BYTE READY, T
86 0010DA 67C8                      BEQ    RETURN              RETURN IF NO BYTE
87 0010DC 14F900018005              MOVE.B ADLC+5,(A2)+        STORE BYTE
88 0010E2 4E75                      RTS                        RETURN


90                                  END
91
92                          *©

****** TOTAL ERRORS   0--   0


SYMBOL TABLE

ADLC       018000 CLEAR      001056 DATA       003000 ENDFLG     001500
FIREUP     001028 GOTFRM     0010C0 LSBT3      01880F MORE       0010A6
MSBT3      01880D PROCES     001074 RECV       0010AE RECVBF     002000
RETURN     0010A4 SEND       001086 START      001000 TRYINP     0010D2
WCR13      018801 WCR2       018803
```

Figure 8. Program Listing (Concluded)